MPT Startup Failures: Workarounds

Occasionally, PBS jobs may fail due to mpiexec timeout problems at startup. If this happens, you will typically see several error messages similar to the following:

- MPT: xmpi_net_accept_timeo/accept() timeout
- MPT ERROR: could not run executable.

When mpiexec starts, several things must happen in a timely manner before the job runs; for example, every node must access the executable and load it into memory. In most cases, this process is successful and the job starts normally. However, if something slows down this process, mpiexec may timeout before the job can start.

This can happen when the job is running an executable located on a filesystem that is not mounted during PBS prologue activities (for example, on a colleague's filesystem); mpiexec may timeout before the filesystem is mounted on each node. (Jobs running on a large number of nodes may be more likely to experience this problem than smaller jobs.)

Workaround Steps

If you encounter an mpiexec startup failure, complete all of these steps to try and resolve the problem.

1. Ensure all the filesystems are mounted before mpiexec runs. For example, if your home filesystem is /home3, your nobackup filesystem is /nobackupp2, and the executable is under /nobackupnfs2, you would add the following line to your job script:

   ```
   #PBS -l site=needed=/home3+/nobackupp2+/nobackupnfs2
   ```

2. Even if all the filesystems are mounted, the filesystem servers might be slow to respond to requests to load the executable. To address this, increase the mpiexec timeout period from 20 seconds (default) to 40 seconds by setting value of the MPI_LAUNCH_TIMEOUT environment variable to 40:

   For csh/tsch
   ```
   setenv MPI_LAUNCH_TIMEOUT 40
   ```
   For bash/sh/ksh
   ```
   export MPI_LAUNCH_TIMEOUT=40
   ```

3. Finally, it might take several attempts to ensure that the executable is loaded into memory. You can accomplish this by adding the several_tries script to the beginning of your mpiexec command line:

   ```
   /u/scicon/tools/bin/several_tries mpiexec -np 2000 /your_executable
   ```

   The script will then attempt the mpiexec command several times until command succeeds (or runs longer than the maximum time set in the environment variables), or a threshold number of attempts is exceeded (see several_tries settings, below.

Alternative to Step 3: If it is difficult to add the several_tries tool to your job script—for example, if you have a complicated set of nested scripts and it's not clear where to insert the tool—try using the pdsh command instead, as follows:

```
pdsh -F $PBS_NODEFILE -a "md5sum /your_executable" | dshbak -c
```

This does a parallel SSH into each node and loads the executable into each node's memory so that it's available when mpiexec tries to launch it.

If the executable is already in your PATH, you do not need to include the entire path in the command line.

Sample PBS Script with Workaround

The following sample script incorporates all three steps described in the previous section.

```
#PBS -l select=200:ncpus=40:model=sky_ele
#PBS -l walltime=HH:MM:SS
#PBS -l site=needed=/home3+/nobackupp2+/nobackupp8+/nobackupnfs2
#PBS -j oe

cd $PBS_O_WORKDIR

setenv MPI_LAUNCH_TIMEOUT 40
set path = ($path /u/scicon/tools/bin)

several_tries mpiexec -np 8000 ./my_a.out
```

several_tries Settings

The following environment variables are associated with the several_tries script:

SEVERAL_TRIES_MAXTIME
: Maximum time the command can run each time, and still be retried.
: Default: 60 seconds

SEVERAL_TRIES_NTRIES

Threshold number of attempts. Default: 6

SEVERAL_TRIES_SLEEPTIME

Sleep time between attempts. Default: 30 seconds

Article ID: 526
Last updated: 21 Dec, 2018
Updated by: Moyer M.
Revision: 11
Running Jobs with PBS -> Optimizing/Troubleshooting -> MPT Startup Failures: Workarounds
https://www.nas.nasa.gov/hecc/support/kb/entry/526/